IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Original) A method for managing a scene graph, the method comprising:

determining a current location for a viewpoint;

determining a current locality threshold based on at least the current location of the viewpoint;

determining which portions of the scene graph are relevant to the current locality threshold;

loading into a local memory those portions of the scene graph that are relevant within the current locality threshold; and

replacing portions of the scene graph that are not relevant within the current locality threshold with one or more pointers, wherein the pointers indicate where the replaced portions may be loaded from if the replaced portions are needed.

2. (Original) The method of claim 1, further comprising:

determining a predicted future locality threshold; and

loading into the local memory those portions of the scene graph that are relevant within the predicted future locality threshold, wherein said replacing is performed only on portions of the scene graph that are not relevant within (i) the current locality threshold and (ii) the predicted future locality threshold.

3. (Original) The method of claim 2, further comprising:

determining a current velocity for the viewpoint, wherein the predicted future locality threshold is determined based on at least the current location for the viewpoint and the current velocity of the viewpoint.

4. (Original) The method of claim 3, wherein said velocity comprises both translational and rotational components.

5. (Original) The method of claim 3, further comprising:

determining a current acceleration for the viewpoint, wherein the predicted future locality threshold is determined based on at least the following:

the current location for the viewpoint,

the current velocity of the viewpoint, and

the current acceleration of the viewpoint.

6. (Original) The method of claim 5, wherein said acceleration comprises both translational and rotational components.

7. (Original) The method of claim 3, further comprising:

determining a current orientation for the viewpoint, wherein the predicted future locality threshold is determined based on at least the following:

the current location for the viewpoint,

the current velocity of the viewpoint, and

the current orientation of the viewpoint.

8. (Original) The method of claim 3, further comprising caching the replaced portions of the scene graph to a local hard drive, wherein the pointers point the replaced portions on the local hard drive.

9. (Original) The method of claim 3, wherein the pointers point to network locations from which the replaced portions of the scene graph may be loaded.

10. (Original) The method of claim 3, further comprising:

compressing the replaced portions of the scene graph; and

storing the compressed portions of the scene graph, wherein the pointers indicate where the compression portions of the scene graph are stored.

11. (Original) The method of claim 1, wherein said replacing is performed only once a predetermined level of memory utilization is reached.

12. (Original) The method of claim 1, wherein said replacing employs hysteresis to prevent thrashing.

13. (Withdrawn) A method for managing a scene graph, the method comprising:

determining a current viewpoint for viewing a virtual world of objects described by the scene graph;

determining a predicted future viewpoint for viewing the virtual world;

determining which of the objects in the virtual world are not visible from the current viewpoint and the predicted future viewpoint; and

replacing portions of the scene graph with pointers, wherein the replaced portions correspond to the objects that are not visible from the current viewpoint and the predicted future viewpoint, and wherein the pointers indicate where the replaced portions may be loaded from.

14. (Withdrawn) The method of claim 13, further comprising:

loading any portions of the scene graph that were replaced by pointers and that correspond to objects that are visible from the current viewpoint or the predicted future viewpoint; and

removing the pointers that correspond to the loaded portions of the scene graph.

15. (Withdrawn) The method of claim 13, further comprising determining a current velocity for the current viewpoint, wherein said predicted future locality threshold volume is calculated based on at least the current viewpoint and the current velocity.

16. (Withdrawn) The method of claim 15, wherein said velocity comprises both translational and rotational components.

17. (Withdrawn) The method of claim 13, further comprising determining a current acceleration for the current viewpoint, wherein said predicted future locality threshold volume is calculated based on at least the current viewpoint and the current acceleration.

18. (Withdrawn) The method of claim 17, wherein said acceleration comprises both translational and rotational components.

19. (Withdrawn) The method of claim 13, further comprising:
compressing the replaced portions of the scene graph; and
storing the compressed portions of the scene graph, wherein the pointers indicate where the compression portions of the scene graph are stored.

20. (Withdrawn) A method for managing a scene graph comprising a plurality of graphical objects, the method comprising:
(a) determining a current viewpoint;
(b) determining a predicted future viewpoint;
(c) determining which of the objects are more than a predetermined threshold distance from the current viewpoint and the predicted future viewpoint; and
(d) replacing portions of the scene graph with pointers, wherein the replaced portions correspond to objects that are more than the predetermined threshold distance from the current viewpoint and the predicted future viewpoint, and wherein the pointers indicate where the replaced portions of the scene graph may be loaded from.

21. (Withdrawn) The method of claim 20, further comprising:
caching the replaced portions of the scene graph to storage locations pointed to by the corresponding pointers.

22. (Withdrawn) The method of claim 20, further comprising:
compressing the replaced portions of the scene graph; and
storing the compressed portions of the scene graph.

23. (Withdrawn) The method of claim 20, further comprising:

repeating (a), (b), (c), and (d) as the viewpoint moves or the scene graph changes; and

restoring the replaced portions of the scene graph in response to one or more objects corresponding to the replaced portions being within the predetermined threshold distance from either the current viewpoint or the predicted future viewpoint.

24. (Withdrawn) The method of claim 20, further comprising:

determining a current velocity for the current viewpoint; and

using the current velocity to determine the predicted future v

25. (Withdrawn) The method of claim 20, further comprising:

determining a current acceleration for the current viewpoint; and

using the acceleration value to determine the predicted future locality threshold.

26. (Withdrawn) The method of claim 20, wherein the replaced portions of the scene graph may comprise visible and non-visible objects.

27. (Withdrawn) A method for managing a scene graph comprising a plurality of graphical objects, the method comprising:

(a) determining a current viewpoint;

(b) determining a predicted future viewpoint;

(c) determining which of the objects are more than a predetermined threshold distance from the current viewpoint and the predicted future viewpoint; and

(d) replacing one or more of the objects of the scene graph with pointers, wherein the replaced objects are more than the predetermined threshold distance from the current viewpoint and the predicted future viewpoint, and wherein the pointers indicate where the replaced objects of the scene graph may be loaded from.

28. (Withdrawn) The method of claim 27, further comprising:

caching the replaced objects of the scene graph to storage locations pointed to by the corresponding pointers.

29. (Withdrawn) The method of claim 27, further comprising:
compressing the replaced objects of the scene graph; and
storing the compressed objects of the scene graph.

30. (Withdrawn) The method of claim 27, further comprising:
repeating (a), (b), (c), and (d) as the viewpoint moves or the scene graph changes; and
restoring the replaced objects of the scene graph in response to one or more of the objects being within the predetermined threshold distance from either the current viewpoint or the predicted future viewpoint.

31. (Withdrawn) The method of claim 27, further comprising:
determining a current velocity for the current viewpoint; and
using the current velocity to determine the predicted future viewpoint.

32. (Withdrawn) The method of claim 27, further comprising:
determining a current acceleration for the current viewpoint; and
using the acceleration value to determine the predicted future viewpoint.

33. (Original) A computer program embodied on a computer-readable medium, wherein the computer program comprises a plurality of instructions that are configured to:
determine a current location for a viewpoint;
determine a current locality threshold based on at least the current location of the viewpoint;
determine which portions of the scene graph are relevant to the current locality threshold;
load into a local memory those portions of the scene graph that are relevant within the current locality threshold; and
replace portions of the scene graph that are not relevant within the current locality threshold with one or more pointers, wherein the pointers indicate where the replaced portions may be loaded from if the replaced portions are needed.

34. (Currently Amended) The computer program of claim 33 [[27]], wherein the instructions are further configured to:
determining a current acceleration; and
using the acceleration value to determine the predicted future viewpoint.

35. (Currently Amended) The computer program of claim 33 [[27]], wherein the instructions are further configured to: render one or more frames based on the scene graph.

36. (Currently Amended) The computer program of claim 33 [[27]], wherein the instructions are further configured to: receive user input regarding movement of the viewpoint.

37. (Currently Amended) The computer program of claim 33 [[27]], wherein the computer program is an application programming interface (API).

38. (Currently Amended) The computer program of claim 33 [[27]], wherein the computer program is a graphics application.

39. (Original) A method for managing a scene graph comprising a plurality of pointers, the method comprising:
determining a current location for a viewpoint;
determining a current locality threshold based on at least the current location of the viewpoint;
determining which of the pointers point to data that is relevant to the current locality threshold;
loading into a local memory the data that is relevant to the current locality threshold;
moving the data that is not relevant to the current locality threshold from the local memory to a new location; and

redirecting pointers in the scene graph that correspond to the moved data to point to the new location.

40. (Original) The method of claim 39, wherein the current locality threshold equals a current view frustum.

41. (Original) The method of claim 39, wherein the moving is only performed if the local memory reaches a predetermined level of fullness.

42. (New) The method of claim 1, wherein said determining which portions of the scene graph are relevant within the current locality threshold comprises determining which of the objects in the scene graph are visible from the current viewpoint location.

43. (New) The method of claim 1, wherein the replaced portions of the scene graph correspond to objects in the scene graph that are not visible from the current viewpoint location.